MEGA Technical Manual

Michael Hillman* Edouard Yreux Kuan Chung Lin†
Guohua Zhou‡

June 6, 2022

1 Historical Technical Background

The Meshfree Explicit Galerkin Analysis (MEGA) program is a state of the art nonlinear computational solid mechanics program for research purposes. The origins trace back to the time Edouard Yreux and Mike Hillman worked under J.S. Chen at UCSD (the pioneer of meshfree methods), with discussions of writing a program based on the advancements in Galerkin meshfree methods at the time, which, until recently, were not implemented into a commercial code. As of today, MEGA is a code for the research group of Mike Hillman and has been used in several projects.

2 Introduction

The MEGA program is an explicit dynamic finite-deformation meshfree code with nonlinear hypo-elastic-plastic constitutive models. It is written using FORTAN90 and OpenMP (shared memory) and is thus amenable to large single-node calculations. A thermo-mechanical version (with hyper-elastic, visco-plastic, thermo-visco-elastic materials) is available but is not described here for brevity.

*Kimball Assistant Professor, Department of Civil and Environmental Engineering, The Pennsylvania State University, University Park, PA 16802, mhillman@psu.edu

†Graduate Student Researcher, Department of Civil and Environmental Engineering, The Pennsylvania State University, University Park, PA 16802

‡Postdoctoral Scholar, Department of Civil and Environmental Engineering, The Pennsylvania State University, University Park, PA 16802

3 Background

3.1 Governing Equations for small deformations

First, small deformations are employed here to introduce some basic concepts. The boundary value problem for infinitesimal deformations is given below

$$\nabla \cdot \boldsymbol{\sigma} + \boldsymbol{b} = \boldsymbol{0} \quad \text{in } \Omega \tag{1a}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{h} \quad \text{on } \partial\Omega_h \tag{1b}$$

$$\boldsymbol{u} = \boldsymbol{g} \quad \text{on } \partial\Omega_g. \tag{1c}$$

Here $\boldsymbol{u}$ is the displacement field, $\boldsymbol{\sigma} = \mathbb{C} : \nabla^{\mathrm{s}}\boldsymbol{u}$ is the Cauchy stress tensor, where $\mathbb{C}$ is the elasticity tensor and $\nabla^{\mathrm{s}}\boldsymbol{u} = 1/2(\nabla \otimes \boldsymbol{u} + \boldsymbol{u} \otimes \nabla)$ is the strain tensor.

3.2 Weak form for small deformations

The weak form of the BVP seeks $\boldsymbol{u} \in U_g$, $U_g = \{\boldsymbol{u} | \boldsymbol{u} \in H^1(\Omega), \boldsymbol{u} = \boldsymbol{g} \text{ on } \partial\Omega_g\}$ such that for all $\boldsymbol{v} \in V_0$, $V_0 = \{\boldsymbol{v} | \boldsymbol{v} \in H^1(\Omega), \boldsymbol{v} = \boldsymbol{0} \text{ on } \partial\Omega_g\}$ the following equation holds:

$$a(\boldsymbol{v}, \boldsymbol{u})_\Omega = (\boldsymbol{v}, \boldsymbol{b})_\Omega + (\boldsymbol{v}, \boldsymbol{h})_{\partial\Omega_h} \tag{2}$$

where

$$a(\boldsymbol{v}, \boldsymbol{u})_\Omega = \int_\Omega \nabla^{\mathrm{s}}\boldsymbol{v} : \mathbb{C} : \nabla^{\mathrm{s}}\boldsymbol{u} \, \mathrm{d}\Omega \tag{3a}$$

$$(\boldsymbol{v}, \boldsymbol{b})_\Omega = \int_\Omega \boldsymbol{v} \cdot \boldsymbol{b} \, \mathrm{d}\Omega \tag{3b}$$

$$(\boldsymbol{v}, \boldsymbol{h})_{\partial\Omega_h} = \int_{\partial\Omega_h} \boldsymbol{v} \cdot \boldsymbol{h} \, \mathrm{d}\Gamma \tag{3c}$$

The Galerkin approximation seeks $\boldsymbol{u} \in U^h$ such that for all $\boldsymbol{v} \in V^h$

$$a(\boldsymbol{v}^h, \boldsymbol{u}^h)_\Omega = (\boldsymbol{v}^h, \boldsymbol{b})_\Omega + (\boldsymbol{v}^h, \boldsymbol{h})_{\partial\Omega_h} \tag{4}$$

where $U^h \in U_g$ and $V^h \in V_0$.

3.3 Reproducing Kernel Approximation

Let a domain $\bar{\Omega} = \Omega \cup \partial\Omega$ be discretized by a set of Np nodes $\mathcal{N} = \{x_1, \cdots, x_{N_P} | x_I \in \bar{\Omega}\}$ with corresponding node numbers $\mathcal{Z} = \{I | x_I \in \mathcal{Z}\}$. The n^{th} order reproducing kernel (RK) approximation $u^h(\boldsymbol{x})$ of a function $u(\boldsymbol{x})$ is [5, 12]:

$$u^h(\boldsymbol{x}) = \sum_{I \in \mathcal{Z}} \Psi_I(\boldsymbol{x}) u_I \tag{5}$$

where $\{\Psi_I(\boldsymbol{x})\}_{I \in \mathcal{Z}}$ is the set of RK shape functions, and $\{u_I\}_{I \in \mathcal{Z}}$ are the associated coefficients. In the solution of PDEs, the coefficients are those of the test and trial functions.

The shape functions are constructed by the product of a kernel function $\Phi_a(\boldsymbol{x}-\boldsymbol{x}_I)$ and a correction function $C(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_I)$:

$$\Psi_I(\boldsymbol{x}) = \Phi_a(\boldsymbol{x}-\boldsymbol{x}_I)C(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_I). \tag{6}$$

The correction function is composed of bases and associated coefficients, which allows the exact reproduction of the bases contained within. The function can be written as a column vector $\boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}_I)$ consisting of complete n^{th} order monomials and a column vector of coefficients $\boldsymbol{b}(\boldsymbol{x})$:

$$C(\boldsymbol{x};\boldsymbol{x}-\boldsymbol{x}_I) = \boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}_I)^T\boldsymbol{b}(\boldsymbol{x}). \tag{7}$$

The coefficients $\boldsymbol{b}(\boldsymbol{x})$ are obtained by enforcing the following monomial reproducing conditions:

$$\sum_{I\in\mathcal{Z}} \Psi_I(\boldsymbol{x})\boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}_I) = \boldsymbol{H}(\boldsymbol{0}). \tag{8}$$

Employing (14)-(16), the RK shape functions are constructed as:

$$\Psi_I(\boldsymbol{x}) = \boldsymbol{H}(\boldsymbol{0})\boldsymbol{M}(\boldsymbol{x})^{-1}\boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}_I)\Phi_a(\boldsymbol{x}-\boldsymbol{x}_I) \tag{9}$$

where

$$\boldsymbol{M}(\boldsymbol{x}) = \sum_{I\in\mathcal{Z}} \boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}_I)\boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}_I)^T\Phi_a(\boldsymbol{x}-\boldsymbol{x}_I) \tag{10}$$

and is called the moment matrix. The MLS approximation can be viewed as a subset of the RK approximation, as the construction of $u^h(\boldsymbol{x})$ by (18) with monomial bases co-insides with the construction of the MLS approximation.

4 Large Deformation Formulation

Galerkin meshfree methods offer a path forward to compute solutions to extremely large deformation problems where the traditional finite element approach is generally ineffective [3]. A meshfree discretization is simply a point-cloud of nodes and surface information, and constructs shape functions directly in Cartesian coordinates. Thus, a meshfree approach and does not rely on a mesh to construct approximation functions. Correspondingly, these methods offer great advantages in extremely large deformation problems where Lagrangian finite elements become distorted or entangled. This is made possibly by the fact that node connectivity and shape functions can be continually reconstructed, rather than being dictated by a Lagrangian element topology. The Meshfree Explicit Galerkin Analysis (MEGA) code is an explicit, large-strain nonlinear dynamic code based on the Reproducing Kernel Particle Method (RKPM) [5, 12]. The reproducing kernel approximation is employed for test and trial functions resulting in a Bubnov-Galerkin formulation discretization in space under the Updated Lagrangian formulation [1]. In order to handle extreme deformations where the mapping between undeformed configuration and current configuration is no longer one-to-one (for instance when free surface formation occurs,

as in simulation of extreme events), a semi-Lagrangian RKPM discretization [8] is employed which constructs the meshfree approximations in the current configuration, all calculations are performed in the current configuration.

Nodal integration is employed for (spatial) domain integration; meaning that the meshfree particles (or nodes) are used as the integration points themselves. Thus, the meshfree particles serve as Lagrangian material points, allowing natural treatment of path-dependent material models. A stain-smoothing method called Stabilized Non-conforming Nodal Integration (SNNI) [8] is adopted in MEGA in order to remedy the instability inherent in nodal integration, and to provide improved solution accuracy over direct nodal integration. Naturally Stabilized Nodal Integration [9] is further introduced under the stain-smoothing framework, which provides additional coercivity in nodal integration.

With the spatial discretization in hand, time-space calculations are performed using the Newmark-Beta time integration scheme [10] with the central difference method. To make the algorithm fully explicit, a lumped mass matrix is utilized, along with mass-proportional damping, resulting in row equations.

5 Spatial Discretization

5.1 Updated Lagrangian Scheme

Let $\Omega_{\boldsymbol{X}}$ denote the the initial configuration of the body with material coordinates $\boldsymbol{X}$ and boundary $\partial\Omega_{\boldsymbol{X}}$, and let $\Omega_{\boldsymbol{x}}$ denote the the current configuration of the body with current coordinates $\boldsymbol{x}$ and boundary $\partial\Omega_{\boldsymbol{x}}$ at time t. The updated Lagrangian equation of motion can be derived using the principle of virtual power in the current configuration [1, 2]:

$$\int_{\Omega_{\boldsymbol{x}}} \boldsymbol{\nabla}_{\boldsymbol{x}}\delta\boldsymbol{v} : \sigma d\boldsymbol{x} - \int_{\Omega_{\boldsymbol{x}}} \delta\boldsymbol{v}\cdot\boldsymbol{b}\rho d\boldsymbol{x} - \int_{\partial\Omega^h_{\boldsymbol{x}}} \delta\boldsymbol{v}\cdot\boldsymbol{h} d\boldsymbol{x} + \int_{\Omega_{\boldsymbol{x}}} \delta\boldsymbol{v}\cdot\dot{\boldsymbol{v}}\rho d\boldsymbol{x} = 0. \quad (11)$$

where δ is the variational operator, $\boldsymbol{\nabla}_{\boldsymbol{x}}$ is the left gradient with respect to the current coordinates $\boldsymbol{x}$, $\boldsymbol{v}$ is the material velocity, σ is the Cauchy stress, $\boldsymbol{b}$ is the prescribed body force in the current configuration, ρ is the density in the current configuration, $\dot{(\,)}$ denotes differentiation with respect to time, and $\boldsymbol{h}$ is the prescribed traction on boundary of the body in the current configuration $\partial\Omega^h_{\boldsymbol{x}}$.

5.2 Semi-Lagrangian Reproducing Kernel Approximation

Let the domain $\bar{\Omega}_{\boldsymbol{x}} = \Omega_{\boldsymbol{x}} \cup \partial\Omega_{\boldsymbol{x}}$ be discretized by a set of Np nodes $\mathcal{N} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{N_P} | \boldsymbol{x}_I \in \bar{\Omega}_{\boldsymbol{x}}\}$ with corresponding node numbers $\mathcal{Z} = \{I | \boldsymbol{x}_I \in \mathcal{N}\}$. In the semi-Lagrangian scheme [8], the nodal locations in the current configuration follow the motion of the body, i.e., $\boldsymbol{x}_I = \boldsymbol{x}(\boldsymbol{X}_I, t)$ where $\boldsymbol{X}_I$ are nodal locations in the undeformed configuration, but shape functions are constructed with respect to distance measures in the current configuration.

In conjunction with the Updated Lagrangian Scheme, this avoids any mapping between the current and undeformed configuration, which is invalid in the presence of extremely large deformations. A typical example is when free surface formulation or closure occurs and the mapping is no longer one-to-one.

Thus the employment of these two formulations enables MEGA to simulate problems such as natural and man-made disasters.

The n^{th} order semi-Lagrangian reproducing kernel (RK) approximation of the displacement field $d(\boldsymbol{x}, t)$ is constructed as [8]:

$$\boldsymbol{d}^h(\boldsymbol{x}, t) = \sum_{I \in \mathcal{Z}} \Psi_I(\boldsymbol{x}) \boldsymbol{d}_I(t) \tag{12}$$

where $\{\Psi_I(\boldsymbol{x})\}_{I \in \mathcal{Z}}$ is the set of RK shape functions, and $\{\boldsymbol{d}_I(t)\}_{I \in \mathcal{Z}}$ are the associated coefficients. It is important to note, that in general the RK shape function lacks the Kronecker delta property ($\Psi_I(\boldsymbol{x}_J) \neq \delta_{IJ}$), and thus the coefficients are not the actual values of displacements at the nodes, and are termed *generalized displacements*. This also results in difficulty in imposing boundary conditions. In MEGA, the boundary singular kernel technique [6] is adopted where nodal coefficients on the boundary take on the value of their associated field variables, and essential boundary conditions are enforced directly at the nodes.

The shape functions are constructed by the product of a kernel function $\Phi_a(\boldsymbol{x} - \boldsymbol{x}_I)$ and a correction function $C(\boldsymbol{x}; \boldsymbol{x} - \boldsymbol{x}_I)$:

$$\Psi_I(\boldsymbol{x}) = \Phi_a(\boldsymbol{x} - \boldsymbol{x}_I) C(\boldsymbol{x}; \boldsymbol{x} - \boldsymbol{x}_I) \tag{13}$$

where

$$C(\boldsymbol{x}; \boldsymbol{x} - \boldsymbol{x}_I) = \boldsymbol{H}(\boldsymbol{x} - \boldsymbol{x}_I)^T \boldsymbol{b}(\boldsymbol{x}). \tag{14}$$

In the above, $\boldsymbol{H}(\boldsymbol{x} - \boldsymbol{x}_I)$ is a column vector consisting of complete n^{th} order monomials, and $\boldsymbol{b}(\boldsymbol{x})$ is a column vector of coefficients. For example, for $d = 2$ and $n = 2$:

$$\boldsymbol{H}(\boldsymbol{x} - \boldsymbol{x}_I) = [1, x, y, x^2, xy, y^2]^T. \tag{15}$$

The kernel function $\Phi_a(\boldsymbol{x} - \boldsymbol{x}_I)$ defines the locality of the approximation, and also the smoothness. For instance, a C^2 cubic B-spline kernel gives C^2 continuity of the approximation. In MEGA, a variety of kernel function are available with different levels of smoothness. The coefficients $\boldsymbol{b}(\boldsymbol{x})$ enforce the monomial reproducing conditions:

$$\sum_{I \in \mathcal{Z}} \Psi_I(\boldsymbol{x}) \boldsymbol{H}(\boldsymbol{x}_I) = \boldsymbol{H}(\boldsymbol{x}). \tag{16}$$

or equivalently,

$$\sum_{I \in \mathcal{Z}} \Psi_I(\boldsymbol{x}) \boldsymbol{H}(\boldsymbol{x} - \boldsymbol{x}_I) = \boldsymbol{H}(\boldsymbol{0}). \tag{17}$$

With the coefficients obtained from (16), the RK shape functions are constructed as:

$$\Psi_I(\boldsymbol{x}) = \boldsymbol{H}(\boldsymbol{0}) \boldsymbol{M}(\boldsymbol{x})^{-1} \boldsymbol{H}(\boldsymbol{x} - \boldsymbol{x}_I) \Phi_a(\boldsymbol{x} - \boldsymbol{x}_I). \tag{18}$$

where

$$\boldsymbol{M}(\boldsymbol{x}) = \sum_{I \in \mathcal{Z}} \boldsymbol{H}(\boldsymbol{x} - \boldsymbol{x}_I)\boldsymbol{H}(\boldsymbol{x} - \boldsymbol{x}_I)^T \Phi_a(\boldsymbol{x} - \boldsymbol{x}_I). \tag{19}$$

In (19), the moment matrix needs to be invertable in order to be able to construct the RK shape functions (18). This requires a minimum number of nodes (with non coplanar locations) covering a given point $\boldsymbol{x}$, which can be difficult to achieve in simulation of extremely large deformations, particularly in fragment-impact problems. To remedy this situation, MEGA employs the quasi-linear RK approximation [13], which guarantees that the moment matrix is never singular. In this approach, when insufficient neighbor coverage is encountered, the order of the approximation reduces.

5.3 Discretization

For discretization of (11), the displacements are approximated by (12), and velocities $\boldsymbol{v}$ and accelerations $\boldsymbol{a}$ are obtained straightforwardly from (12):

$$\boldsymbol{v}^h(\boldsymbol{x}, t) = \sum_{I \in \mathcal{Z}} \Psi_I(\boldsymbol{x})\dot{\boldsymbol{d}}_I(t) \tag{20a}$$

$$\boldsymbol{a}^h(\boldsymbol{x}, t) = \sum_{I \in \mathcal{Z}} \Psi_I(\boldsymbol{x})\ddot{\boldsymbol{d}}_I(t) \tag{20b}$$

After substituting (12) and (20) into (11), and employing the arbitrary nature of the virtual coefficients, we have the following semi-discrete matrix form:

$$\boldsymbol{M}\boldsymbol{a} = \boldsymbol{f}^{ext} - \boldsymbol{f}^{int} \tag{21}$$

where $\boldsymbol{a}$ is the column vector of generalized accelerations $\{\ddot{\boldsymbol{d}}_I\}_{I \in \mathcal{Z}}$, $\boldsymbol{M}$, $\boldsymbol{f}^{ext}$, and $\boldsymbol{f}^{int}$ are the mass matrix, external force, and internal force, respectively:

$$\boldsymbol{M}_{IJ} = \boldsymbol{I}_3 \int_{\Omega_{\boldsymbol{x}}} \Psi_I(\boldsymbol{x})\Psi_J(\boldsymbol{x})\rho d\boldsymbol{x} \tag{22a}$$

$$\boldsymbol{f}_I^{ext} = \int_{\Omega_{\boldsymbol{x}}} \Psi_I(\boldsymbol{x})\boldsymbol{b}(\boldsymbol{x})\rho d\boldsymbol{x} + \int_{\partial\Omega_{\boldsymbol{x}}^h} \Psi_I(\boldsymbol{x})\boldsymbol{t}(\boldsymbol{x})d\boldsymbol{x} \tag{22b}$$

$$\boldsymbol{f}_I^{int} = \int_{\Omega_{\boldsymbol{x}}} \boldsymbol{B}_I^T(\boldsymbol{x})\boldsymbol{\Sigma}(\boldsymbol{x})d\boldsymbol{x} \tag{22c}$$

where $\boldsymbol{I}_3$ is a 3×3 identity matrix, $\boldsymbol{b}(\boldsymbol{x})$ and $\boldsymbol{t}(\boldsymbol{x})$ are the vector forms of the body force and prescribed tractions, respectively, $\boldsymbol{B}^T(\boldsymbol{x})$ is the strain-gradient matrix, and $\boldsymbol{\Sigma}(\boldsymbol{x})$ is the matrix form of the Cauchy stress.

6 Numerical Quadrature in MEGA

RKPM shape functions are rational, and can form complicated overlapping support structures that are difficult to integrate accurately. Because of this, the

rate of convergence of the solution can be heavily influenced by the choice of domain integration.

High-order Gaussian quadrature can yield convergent results, yet is computationally prohibitive. On the other hand, nodal integration is highly efficient and keeps the desirable characteristics of meshfree methods on the discrete level with quadrature, but can yield non-convergent results and solution instability [7]. The former is due to under-integration, while the latter is caused by severely underestimating the energy for saw-tooth modes by using nodes as sampling points. Advanced nodal integration methods that address these issues are implemented in MEGA.

6.1 Stabilized Nodal Integration

A stabilized Conforming Nodal Integration (SCNI) [7] has been introduced in to remedy rank instability in direct nodal integration, and also provide optimal convergence for linearly complete shape functions. In this method, gradients are smoothed over conforming nodal representative domains which partition the domain, so that they are not evaluated directly at the nodes thus avoiding rank instability. Nodal integration is then performed with the smoothed nodal gradients in hand.

The smoothing is also performed in such a way that the first order variational consistency condition (for Galerkin linear exactness) is satisfied. This is the requirement to satisfy the following divergence equality with the set of test functions and the chosen numerical integration [7]:

$$\hat{\int}_{\Omega_{\boldsymbol{x}}} \boldsymbol{\nabla}_{\boldsymbol{x}} \Psi_I(\boldsymbol{x}) \, \mathrm{d}\Omega = \hat{\int}_{\partial\Omega_{\boldsymbol{x}}} \Psi_I(\boldsymbol{x}) \boldsymbol{n}(\boldsymbol{x}) \, \mathrm{d}\Gamma \quad \forall I \tag{23}$$

where " $\hat{}$ " denotes numerical integration, and $\Psi_I(\boldsymbol{x})$ is a shape function with first order completeness used in the Galerkin equation.

SCNI considers gradient smoothing with divergence in each nodal representative domain by

$$\tilde{\boldsymbol{\nabla}}_{\boldsymbol{x}} \Psi_I(\boldsymbol{x}_L) = \frac{1}{|\Omega_L|} \int_{\Omega_L} \boldsymbol{\nabla}_{\boldsymbol{x}} \Psi_I(\boldsymbol{x}) \, \mathrm{d}\Omega = \frac{1}{|\Omega_L|} \int_{\partial\Omega_L} \Psi_I(\boldsymbol{x}) \boldsymbol{n}(\boldsymbol{x}) \, \mathrm{d}\Gamma. \tag{24}$$

Here $|\Omega_L| = \int_{\Omega_L} \mathrm{d}\Omega$ and Ω_L is the representative domain of node L. The conforming nodal domains can be generated by, for example, Voronoi diagrams. When the gradient approximation (24) is employed in the Galerkin equation in conjuction with first-order complete approximations, the integration constraint (23) is satisfied, and exactness in linear problems (passing the linear patch test) is attained, along with optimal convergence rates associated with linear completeness.

The formation of conforming strain smoothing domains in SCNI can be cumbersome in problems subjected to topological change in geometry, and stabilized non-conforming nodal integration (SNNI) [8] has been introduced as a simplification of SCNI. Gradient smoothing schemes by non-conforming cells constructed

by considering box domains surrounding the node can be constructed, and corrected [4]. In MEGA, the smoothing cells are constructed to be conforming in the initial configuration.

To satisfy linear completeness, it is sufficient to take single point integration on each boundary face in the smoothed methods (24) so that the smoothed gradients meet the gradient partition of nullity and the first order gradient reproducing condition, if the approximation is first order complete [4]. Thus in MEGA, since linear basis is most often employed, single-point integration is chosen which can maintain linear completeness in the gradient approximation.

The energy of sawtooth modes may still be under-sampled in smoothed nodal integration methods when the surface area to volume ratio of the domain is relatively small, or when the discretization is sufficiently fine. To this end, a Naturally Stabilized Nodal Integration (NSNI) [9] is implemented into MEGA under the strain-smoothing framework, which provides additional coercivitey in nodal integration and precludes these sawtooth modes.

7 Constitutive modeling

The MEGA code has small deformation elasticity, and finite-strain plasticity models. The elastic model is only valid for small deformations and has not yet been generalized to large strains and rotations, e.g., viz the Neo-Hookean model. Plasticity models in MEGA include $J2$ (Von-Mises) plasticity with isotropic hardening and isotropic damage (e.g., for metals), and Drucker-Prager plasticity with tension cut-off and isotropic damage (e.g., for geomaterials).

7.1 Objective Stress update

MEGA employs the Jaumann rate of the Cauchy stress $\mathring{\sigma}$ to maintain objectivity in constitutive modeling (c.f. [2] for a summary):

$$\begin{aligned} \mathring{\sigma} &= C^J : D \\ \mathring{\sigma} &= \dot{\sigma} - \Omega \cdot \sigma - \sigma \cdot \Omega^T \end{aligned} \tag{25}$$

where Ω is the spin tensor, C^J is the material response tensor in relation to the Jaumann rate, and D is the rate of deformation tensor. For the time step from t^n to t^{n+1}, Hughes and Winget [11] showed that the above can be integrated in time in an *incrementally* objective way (preserving further objectivity in a discrete sense) by first defining an incremental deformation gradient G with respect to to the configuration $\boldsymbol{x}^{n+1/2} = 1/2\left(\boldsymbol{x}^n + \boldsymbol{x}^{n+1}\right)$:

$$G_{ij} = \frac{\partial \Delta d_i}{\partial x_j^{n+1/2}} \tag{26}$$

where $\Delta \boldsymbol{d} = \boldsymbol{d}^{n+1} - \boldsymbol{d}^n$ is the increment of displacement. The gradient (26) is computed using the smoothed shape functions in Section 6.

The symmetric part of the gradient G is then employed for the strain measure in calculating the elastic trial stress in plasticity. With the trial stress σ_{trial}^{n+1} in hand, the true Cauchy stress σ^{n+1} at time $n+1$ is obtained via iteration on the plasticity equations, and the internal force $\boldsymbol{f}_{int}^{n+1}$ is then formed with nodal integration:

$$\boldsymbol{f}_{int}^{n+1} = \sum_{I \in \mathcal{Z}} \tilde{\boldsymbol{B}}^T(\boldsymbol{x}_I) \boldsymbol{\Sigma}^{n+1}(\boldsymbol{x}_I) V_I \tag{27}$$

where $\tilde{\boldsymbol{B}}^T$ and Σ^{n+1} are Voigt notation vectors containing the smoothed spatial gradients and stresses σ^{n+1}, respectively, and V_I is the nodal volume in the current configuration. The contribution to the internal force by NSNI follows analogously and further details can be found in [9].

8 Time integration

Time integration on the semi-discrete form (21) is accomplished using the Newmark-β algorithm with the explicit central difference scheme ($\beta = 0$ and $\gamma = 1/2$ - c.f. [10] for a summary of common time integration methods). First, to make the algorithm fully explicit, the mass in (21) is lumped into a lumped-mass matrix $\boldsymbol{M}^l$ using the row-sum technique:

$$\begin{aligned} \boldsymbol{M}_{IJ}^l &= 0, && I \neq J \\ \boldsymbol{M}_{JJ}^l &= \sum_{I \in \mathcal{Z}} \boldsymbol{M}_{IJ} \quad \text{no sum on } J && \textit{otherwise} \end{aligned} \tag{28}$$

To consider damping in the explicit dynamic context, Rayleigh damping is adopted with purely a mass-proportional damping matrix $\boldsymbol{C}$:

$$\boldsymbol{C} = \alpha \boldsymbol{M}^l \tag{29}$$

where α is an empirically determined coefficient of damping, or can be related to the undamped frequency of vibration and the damping ratio for linear problems.

The predictor-corrector algorithm in *a-form* is employed; given the quantities at the previous timestep n, for time $n+1$, the predictors are first computed:

$$\begin{aligned} \widetilde{\boldsymbol{d}}_{n+1} &= \boldsymbol{d}_n + \Delta t \boldsymbol{v}_n + \frac{\Delta t^2}{2} \boldsymbol{a}_n \\ \widetilde{\boldsymbol{v}}_{n+1} &= \boldsymbol{v}_n + \frac{\Delta t}{2} \boldsymbol{a}_n \end{aligned} \tag{30}$$

Based on the predictors, the shape functions, stresses, and internal forces are updated as described in Sections 5.2 and 7. With the mass lumping and mass-proportional damping in hand, the explicit central difference formulas yield the generalized accelerations at each node:

$$\boldsymbol{a}_I^{n+1} = \frac{1}{M_{II} + \frac{1}{2} C_{II} \Delta t^2} \{ \boldsymbol{f}_{ext}^{n+1}{}_I - \boldsymbol{f}_{int}^{n+1}{}_I - \boldsymbol{C}_{II} \widetilde{\boldsymbol{v}}_{n+1} \} \tag{31}$$

where M_{II} and C_{II} are the the scalar mass and damping coefficient associated with node I from mass lumping (28) and damping equation (29), respectively. The correctors are then calculated as

$$\begin{aligned} \boldsymbol{d}_{n+1} &= \widetilde{\boldsymbol{d}}_{n+1} \\ \boldsymbol{v}_{n+1} &= \widetilde{\boldsymbol{v}}_{n+1} + \frac{\Delta t}{2}\boldsymbol{a}_{n+1} \end{aligned} \tag{32}$$

which begins the process again with $n \leftarrow n+1$ in (31)-(32) for the next time step via the predictors (30).

Acknowledgements We greatly acknowledge the support of this work by the L. Robert and Mary L. Kimball Early Career Professorship.

References

[1] Klaus-Jürgen Bathe. *Finite element procedures*. Klaus-Jurgen Bathe, 2006.

[2] Ted Belytschko, Wing Kam Liu, Brian Moran, and Khalil Elkhodary. *Nonlinear finite elements for continua and structures*. John wiley & sons, 2014.

[3] J. S. Chen, Michael Hillman, and Sheng Wei Chi. Meshfree Methods: Progress Made after 20 Years. *Journal of Engineering Mechanics*, 2016.

[4] Jiun-Shyan Chen, Michael Hillman, and Marcus Rüter. An arbitrary order variationally consistent integration for galerkin meshfree methods. *International Journal for Numerical Methods in Engineering*, 95(5):387–418, 2013.

[5] Jiun Shyan Chen, Chunhui Pan, Cheng-Tang Wu, and Wing Kam Liu. Reproducing Kernel Particle Methods for large deformation analysis of nonlinear structures. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):195–227, 1996.

[6] Jiun-Shyan Chen and Hui-Ping Wang. New boundary condition treatments in meshfree computation of contact problems. *Computer methods in applied mechanics and engineering*, 187(3-4):441–468, 2000.

[7] Jiun-Shyan Chen, Cheng-Tang Wu, Sangpil Yoon, and Yang You. A stabilized conforming nodal integration for galerkin mesh-free methods. *International journal for numerical methods in engineering*, 50(2):435–466, 2001.

[8] Pai Chen Guan, Jiun Shyan Chen, Y. Wu, H. Teng, J. Gaidos, K. Hofstetter, and M. Alsaleh. Semi-Lagrangian reproducing kernel formulation and application to modeling earth moving operations. *Mechanics of Materials*, 41(6):670–683, 2009.

[9] Michael Hillman and Jiun-Shyan Chen. An accelerated, convergent, and stable nodal integration in galerkin meshfree methods for linear and nonlinear mechanics. *International Journal for Numerical Methods in Engineering*, 107(7):603–630, 2016.

[10] Thomas JR Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.

[11] Thomas JR Hughes and James Winget. Finite rotation effects in numerical integration of rate constitutive equations arising in large-deformation analysis. *International journal for numerical methods in engineering*, 15(12):1862–1867, 1980.

[12] Wing Kam Liu, Sukky Jun, and Yi Fei Zhang. Reproducing kernel particle methods, 1995.

[13] Edouard Yreux and Jiun-Shyan Chen. A quasi-linear reproducing kernel particle method. *International Journal for Numerical Methods in Engineering*, 109(7):1045–1064, 2017.